


# Anaconda と Jupyter Notebook を用いた Python プログラミング (4)

※このドキュメントは相互参照型のハイパーテキストを意図しており、Web サイトおよびブックマーク、図表、数式、補足説明へのリンク (参照) を [下線付き青文字](#) で表し、(↪) の記号は初出の参照元へのリンクを表す。



## 4. 粉末 X 線回折データの読み込みと描画, 逆畳込的处理


[4 章](#)と [5 章](#), [6 章](#)では, Anaconda3 と Jupyter Notebook  を用いて, 粉末 X 線回折 (X-ray diffraction; XRD) 測定装置の角度校正<sup>こうせい</sup>を行い, 実測の粉末 XRD データから正確な回折ピーク位置を求めることを目標とした作業を行う。

粉末 X 線回折装置の角度校正<sup>こうせい</sup> (calibration) の目的では, 米国国立標準技術研究所 (National Institute of Standards and Technology; NIST) から頒布<sup>はんぷ</sup>される Si 標準参照物質 (standard reference material) SRM640 シリーズ, あるいは LaB<sub>6</sub> 標準参照物質 SRM660 シリーズを用いるのが標準的な方法である。(補足 4.A)

### 4-1 標準試料測定データの準備

Si 標準粉末 NIST SRM640d について 10 回の粉末 X 線回折強度走査測定を行った結果を積算して, 回折装置の角度校正を行うためのデータとして用いる。XRD データと一部の Python コードをまとめた圧縮ファイル 20240503MiniFlex.zip へのリンクを Moodle あるいは担当者の web サイト (<https://takashiida.net/education/20240613-ceramic-practice-21/>) から提供する。

実習を進めるためには, この圧縮ファイル “20240503MiniFlex.zip” をユーザーのホームディレクトリ下「ダウンロード」(Downloads) ディレクトリで展開 (解凍) して生成される 「 / Downloads / 20240503MiniFlex」フォルダを所定の位置, 例えば 「 / OneDrive - NITech / Documents / 00jikken」フォルダに移動すれば良い。名古屋工業大学の教育用端末では名工大 Microsoft アカウント経由のクラウド・ストレージ OneDrive を使えるので, 端末上のユーザーのホームディレクトリではなく, OneDrive 下で作業を始めれば, 個人で占有使用<sup>せんゆうしゅう</sup>できるパソコンで作業を継続することが容易になる。(補足 4.1.A)

個人のパソコンに実験データを置くときには, OneDrive ディレクトリ下の「ドキュメント」(Documents) ディレクトリ 「 / OneDrive - NITech / Documents」の下に “00jikken” のようなサブディレクトリを作成し, 例えば 「2024 年 5 月 3 日に MiniFlex という機種名の装置で測定したデータ」であれば “20240503MiniFlex” という名称のサブディレクトリを作成し, その中にまとめるように

すると良い。作業開始時には、たとえば [Figure 4.1.1](#) に示すようなディレクトリ（フォルダ）の構成にする。

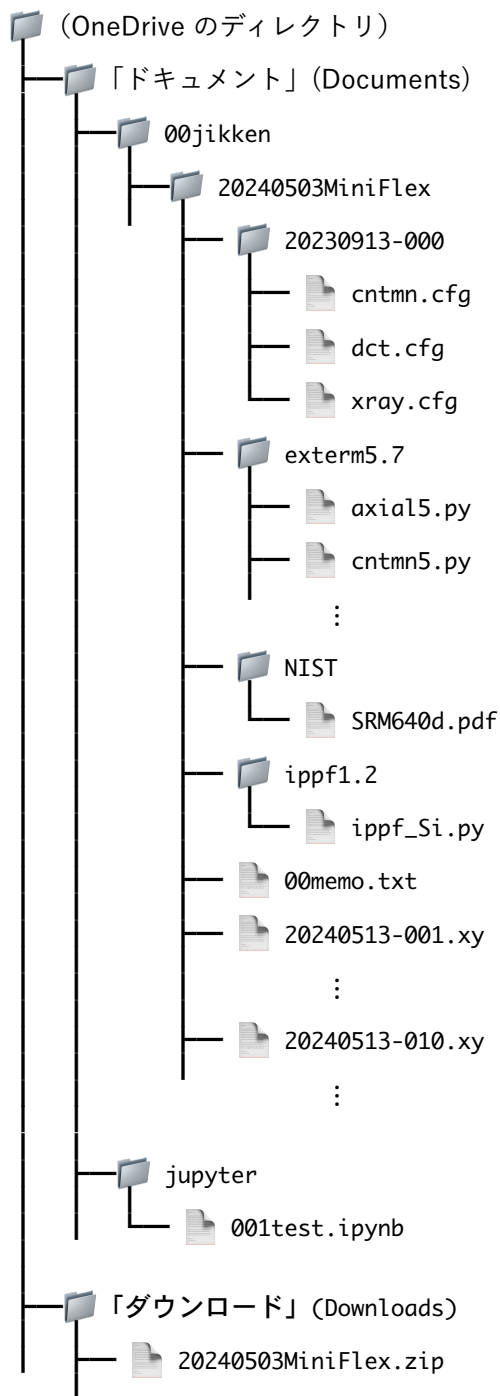





Figure 4.1.1 作業開始時のディレクトリ（フォルダ）とファイル構成例の模式図 (↔)






Windows 11 の標準 web ブラウザ Microsoft Edge  と、ファイル管理システム「エクスプローラー」(File Explorer)  を使って [Figure 4.1.1](#) に示すファイル構成にするためには、例えば、以下のような操作を行えば良い。

- (1) (圧縮ファイルのダウンロード) Microsoft Edge  から担当者の web ページ (<https://takashiida.net/education/20240613-ceramic-practice-21/>) を開き、リンク ([https://takashiida.sadist.jp/public/education/20240613ceramic\\_practice/20240503MiniFlex.zip](https://takashiida.sadist.jp/public/education/20240613ceramic_practice/20240503MiniFlex.zip)) をクリック（タップ）する。

「ダウンロード」ウィンドウの「ファイルを開く」リンクはクリック（タップ）せず、次に進む。



(2) (File Explorer の起動・圧縮ファイルの展開) Windows タスクバーの「エクスプローラー」アイコン  をクリック（タップ）し、「ダウンロード」フォルダを開く。



20240503MiniFlex.zip 圧縮ファイルのアイコンを右クリック（二本指タップ）すると表示されるポップアップメニューから、「すべて展開...」を選択する。「展開先を選んでください。」ダイアログから、目的のフォルダを選択する。（例えば、ローカルディスク C:/Users（ユーザー）/ユーザー名/OneDrive - NITech/Documents/00jikken/ など）


Jupyter Notebook  のファイルブラウザから「 / OneDrive - NITech / Documents / 00jikken /」フォルダに「 20240503MiniFlex /」フォルダが配置され、[Figure 4.1.1](#) に示すようなディレクトリ構成になっていることを確認する。確認できれば「 / Downloads」フォルダ中の“20240503MiniFlex.zip”圧縮ファイルをチェックし、削除  ボタンをクリック（タップ）して削除する。



## 4-2 粉末X線回折データの読み込みと描画


「20240503MiniFlex」フォルダには、NIST Si 標準粉末 SRM640d について、発散スリット角  $1.25^\circ$  設定での 10 回の粉末 X 線回折強度走査測定データを“20240503-001.xy”, ..., “20240503-010.xy” の 10 ファイルとして保存してある。詳細については、測定時に記録として残された“00memo.txt”テキストファイルを参照する。“20240503-\*\*\*.xy”の各ファイルには「カンマ区切り値」(comma-separated values; CSV) 形式でゴニオメーター角度  $2\theta$  と強度値  $Y$  とが記載されている。ただし、各ファイル中の冒頭部 424 行は「ヘッダ」(header) として、測定条件などの情報を記載する設定としてある。

Jupyter Notebook のファイル管理システムを使ってこれらの強度データファイルの内容を確認する。2-1 節に示した Jupyter Notebook ファイルブラウザ (Home タブ) 画面から、([4-1 節](#) に示したディレクトリ構成の場合には) 「 / OneDrive / Documents / 00jikken / 20240503MiniFlex /」フォルダ (ディレクトリ) へ移行する。これ以後は、このフォルダを作業用フォルダとする。ファイルのリスト中に表示される「 20240503-001.xy」をダブルクリック (ダブルタップ) すれば、Web ブラウザ上で新しいタブ“20240503-001.xy ...”が開き、ファイルの内容が表示される。冒頭の 424 行がヘッダ部であることと、角度・強度データの並び方を確認し、このファイル内容表示タブ“20240503-001.xy ...”を閉じる。



以下、個々の強度ファイル「 20240503-001.xy」…「 20240503-010.xy」を読み込み、グラフとして表示するための Python コードを作成する。


Notebook のファイルブラウザ「 / OneDrive / Documents / 00jikken / 20240503MiniFlex」画面ファイルリスト右上の [New ▼] プルダウンメニューから「Notebook」項目を選択する。「Select Kernel」ダイアログが表示された場合は「Python 3 (ipykernel)」を選択する。初期ノートブック画

面“Untitled - Jupyter Notebook”が表示されたら、Fileメニューから「Close and Shutdown Notebook」という意味の項目を選択する。Jupyter Notebookのファイルブラウザ（Homeタブ）から「 Untitled.ipynb」ファイルにチェックを入れ、[Rename]ボタンからファイル名を“02show\_one\_scan.ipynb”と変更する。この操作により「 / OneDrive / Documents / 00jikken / 20240503MiniFlex」ディレクトリに

「 02show\_one\_scan.ipynb」という名称のJupyter Notebookファイルが作成される。

“02show\_one\_scan”ノートブックのコードセルに、(Code 4.2.1)のようなPythonコードを入力し実行する。Pythonコードはタイピング入力すると良いが、「#」記号以下は「コメント」と呼ばれる部分であり、コードの動作には影響ないので入力しなくても良い。ただし一般的には詳細なコメントを添えるほどコードの管理・デバッグ・再利用などをしやすくなる。(補足 4.2.A)

手や指を動かしにくい人はコピー&ペースト入力するのでも構わない。(Code 4.2.1)を実行すればFigure 4.2.1のようなグラフが得られる。(Code 4.2.1)では、このNotebookファイル「 02show\_one\_scan.ipynb」が存在するのと同じディレクトリに「 20230913-001.xy」という名称のファイルが存在することを前提とすることに注意する。(補足 4.2.B)

(Code 4.2.1) CSV ファイルの読み込みと描画 “02show\_one\_scan.ipynb” ()

```
1) # 02show_one_scan.ipynb
2) # 指定 csv ファイルの読み込みと描画
3) import numpy as np # NumPy ライブラリをインポートし, np として参照する
4) import matplotlib.pyplot as plt # matplotlib.pyplot を plt として参照
5) sRaw='20240503-001.xy' # 入力ファイル名
6) nSkip=424 # 入力ファイル冒頭部スキップ行数
7) # CSV ファイルの読み込み...
8) sLoad=np.loadtxt(sRaw,dtype='float',delimiter=',',skiprows=nSkip)
9)     # numpy.loadtxt() (load text) メソッドを利用する
10)     # 入力ファイル名, データ型 (dtype='float') (浮動小数点数)
11)     # 区切り文字 (delimiter=','), スキップ行数 (skiprows=nSkip) の指定
12) x=sLoad[:,0] # sLoad 二次元配列の各行 0 列目を一次元配列 x とする
13) y=sLoad[:,1] # sLoad 二次元配列の各行 1 列目を一次元配列 y とする
14) # グラフの描画...
15) fig,ax = plt.subplots(figsize=(6.4,3.2))
16)     # fig オブジェクトと ax オブジェクトの生成
17) ax.plot(x, y) # x に対する y の折れ線グラフをプロットする
18) ax.xaxis.set_ticks_position('both') # x軸上下に目盛をつける
19) ax.yaxis.set_ticks_position('both') # y軸左右に目盛をつける
20) ax.minorticks_on() # 補助目盛を表示する
21) ax.set_xlabel(r'$2\Theta$ (^\circ)$') # x軸ラベルの設定
22) ax.set_ylabel('Intensity (counts)') # y軸ラベルの設定
23) ax.set_xlim(5,140) # x 軸描画範囲の設定
24) ax.set_ylim(0,) # y 軸描画範囲の設定
25) plt.show() # matplotlib.pyplot.show() メソッドの実行, グラフ表示
```

()

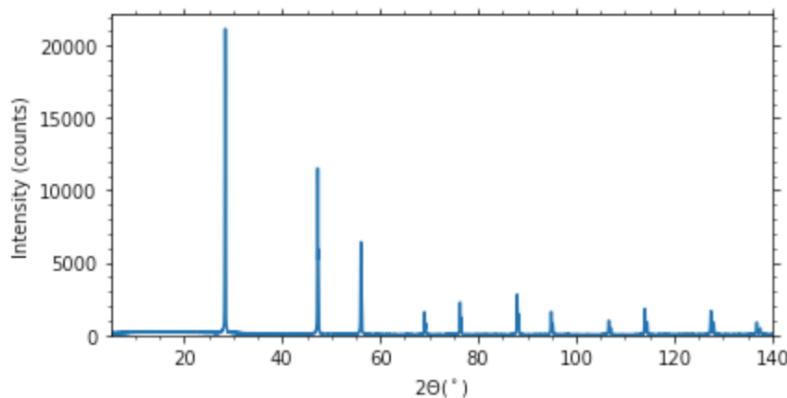


Figure 4.2.1 NIST SRM 640d 標準 Si 粉末試料 XRD 強度データの描画例 (↔)

(Code 4.2.1) 中では、強度データ・ファイルの読み込みに `numpy.loadtxt()` メソッドを用いている。この強度データ・ファイルでは、角度・強度データはゴニオメーター角度（見かけ回折角） $2\theta : 4.38^\circ - 142.47^\circ$  の範囲を  $0.01^\circ$  ステップで測定したものとして記録されている。オブジェクト名 `source` で表される内容は Numpy 配列 (array) 型のオブジェクトであり、ここでは「13811 行 × 2 列」の 2 次元配列（行列）として読み込まれる。

(Code 4.2.1) 中で、グラフの X 軸ラベルを表示する文字列として `r'$2\Theta (^{\circ})$'` と指定している。`r'$...$'` 記号で囲んだ範囲は TeX での表現を用いている。TeX では “\Theta” はギリシャ文字大文字の Θ (Theta) を表し，“^” (hat 記号) は「上付き添字」，“\circ” は「°」(円; circle) を表す。

### 4-3 複数回走査強度データの積算

10 回の粉末 X 線回折走査測定 of 強度データ「`20240503-001.xy`」, ..., 「`20240503-010.xy`」を積算したデータを端末のスクリーン上に描画し、積算によって得られたデータを別名のファイル「`20240503-021.csv`」として保存する。このためには (Code 4.3.1) のような Python コードを実行すれば良い。ノートブックの名称は例えば “03add\_ten\_scans” とする。(補足 4.3.A)

正常に動作する (Code 4.2.1) を持っている状況であれば、タイピング入力に慣れた人であっても、(Code 4.3.1) を入力するために、(Code 4.2.1) の内容をコピー&ペーストして、必要な場所を書き換えるようにすると良い。また書き換える内容の大部分も、元のコードの一部を選択したコピー&ペーストや、必要な箇所の字下げ (indentation) とで済ませる。そのような操作を用いることで、タイピングのミスによる文法エラー (syntax error) の発生を回避しやすくなるだけでなく、プログラミング初心者にとってコードの論理的な構造をむしろ理解しやすくなる面もある。

この Jupyter Notebook ノートブックに記述されたコードは、Notebook ファイル

「`03add_ten_scans.ipynb`」 と強度データファイル「`20240503-001.xy`」, ..., 「`20240503-010.xy`」とが、同じディレクトリ (フォルダ) に存在することを前提とする。一般的なアプリケーション・ソフトウェアとは異なり、コードに限定的な機能しか持たせないことになるが、このノートブックファイル「`03add_ten_scans.ipynb`」を保存しておくことで、積算データ「`20240503-021.csv`」がどのように作成されたかを明確な記録として残しておく意味がある。



「□ 20240503-001.xy」, ..., 「□ 20240503-010.xy」の冒頭部 424 行は、ヘッダ部として読み飛ばす (スキップする) が、積算データ「□ 20240503-021.csv」にはヘッダ部を付けない。

(Code 4.3.1) を走らせて、[Figure 4.3.1](#) のようなグラフが表示されることを確認する。

(Code 4.3.1) 複数ファイルの強度の積算 "03add\_ten\_scans.ipynb" (↔)

```
1) # 03add_ten_scans.ipynb
2) # 複数ファイルの強度の積算
3) import numpy as np
4) import matplotlib.pyplot as plt
5) sSave='20240503-021.csv' # 出力ファイル名
6) # 1 番目のファイルの読み込み...
7) sHead='20240503-' # 入力ファイル名の頭部文字列
8) sBody='{ :03d}'.format(1) # 入力ファイル名の番号部文字列 "001"
9) sTail='.xy' # 入力ファイル名の尾部文字列
10) nSkip=424 # スキップ行数
11) sRaw=sHead+sBody+sTail # 入力ファイル名
12) sLoad=np.loadtxt(sRaw,dtype='float',delimiter=',',skiprows=nSkip)
13) x=sLoad[:,0] # sLoad 二次元配列 (行列) の各行 0 列目を一次元配列 x とする
14) y=sLoad[:,1] # sLoad 二次元配列 (行列) の各行 1 列目を一次元配列 y とする
15) # 2 番目から 10 番目のファイルの読み込みと積算...
16) for i in range(2,11): # 2 から始まり 10 で終わる
17)     sBody='{ :03d}'.format(i) # 入力ファイル名の番号部 "002"- "010"
18)     sRaw=sHead+sBody+sTail # 入力ファイル名
19)     sLoad= np.loadtxt(sRaw,dtype='float',delimiter=',',skiprows=nSkip)
20)     y += sLoad[:,1] # 強度値 y だけ積算する
21) # 積算強度データの出力...
22) source = np.stack([x,y], axis=1) # 列方向 (axis=1) に積んだ二次元配列を作成する
23) np.savetxt(sSave,source,delimiter=',',fmt='% .3f')
24) # グラフの描画...
25) fig, ax = plt.subplots(figsize=(6.4,3.2))
26)     # fig (図) オブジェクトと ax (軸) オブジェクトの生成
27) ax.plot(x,y) # x に対する y の折れ線グラフを表示する
28) ax.xaxis.set_ticks_position('both') # x軸上下に目盛をつける
29) ax.yaxis.set_ticks_position('both') # y軸左右に目盛をつける
30) ax.minorticks_on() # 補助目盛を表示する
31) ax.set_xlabel('$2\Theta$ (^\circ)$') # x軸ラベルの設定
32) ax.set_ylabel('Intensity (counts)') # y軸ラベルの設定
33) ax.set_xlim(5,140) # x軸描画範囲の設定
34) ax.set_ylim(0,) # y軸描画範囲の設定
35) plt.show() # グラフ表示
```

(↔)

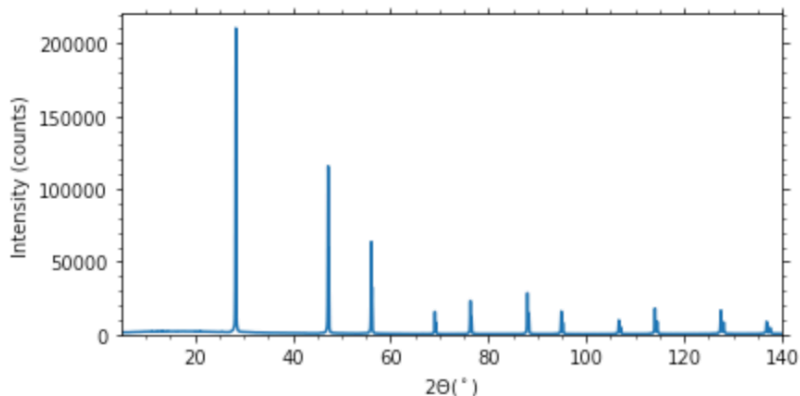


Figure 4.3.1 NIST SRM 640d 標準 Si 粉末試料 10 回走査積算 XRD 強度データの描画例 (↔)




Figure 4.2.1 に示した 1 回走査強度データと、Figure 4.3.1 に示した 10 回走査強度積算データとの相似性と違いとを確認する。

## 4-4 逆畳込的処理の適用

NIST SRM 640d 標準 Si 粉末試料の実測 XRD データに対して、<sup>ぎやくたみこみてきしより デコンボリューション</sup>逆畳込的処理 (deconvolutional <sup>トリートメント</sup>treatment) (Ida, 2020; Ida, 2021) を適用する。この処理によって、実測データに現れている  $\text{CuK}\alpha_2$  <sup>サブピーク</sup>副ピーク (subpeak) や減衰された  $\text{CuK}\beta$  ピーク、Ni K 吸収端位置での不連続な強度変化が除去され、さらに装置取差に由来するピークシフトとピーク形状の非対称な変形が自動的に修正される。(補足 4.4.A) この処理の実装について知りたければ担当者の web サイト (<https://takashiida.sadist.jp/public/education/powder-xrd/710instrumental.pdf>) を参考にすれば良い。

逆畳込的処理を実施するために開発された Python コードは“20240503MiniFlex”ディレクトリ下“extern5.7”フォルダの中に 12 ファイル“axial5.py”, “cntmn5.py”, “common5.py”, “equatorial1.py”, “equatorial2.py”, “equatorial3.py”, “extern5.py”, “transpr1.py”, “transpr2.py”, “transpr3.py”, “transpr4.py”, “xray5.py”として配置してある。このうち“extern5.py”が主プログラムとして機能し、必要に応じて他の 11 の副プログラム“axial.py”, “cntmn5.py”, “common5.py”, “equatorial1.py”, “equatorial2.py”, “equatorial3.py”, “trnspr1.py”, “trnspr2.py”, “trnspr3.py”, “trnspr4.py”, “xray5.py”を呼び出す構成としている。(補足 4.4.B)

4-3 節で作成した 10 回走査強度積算データ“20240503-021.csv”を処理対象とする。

(逆畳込的処理作業の概要) Jupyter Notebook  と Windows のエクスプローラ  とを使って逆畳込的処理を実施する。“20240503MiniFlex”ディレクトリ下に“20240503-021”サブディレクトリを新しく作成し、このサブディレクトリ「 / OneDrive / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」を逆畳込的処理の作業用ディレクトリとする。作業用ディレクトリに、処理対象となる強度データファイル (デフォルトのファイル名“00raw.csv”) と設定 (configuration) <sup>コンフィギュレーション</sup>ファイル“dct.cfg”, “xray.cfg”, “cntmn.cfg”とを配置する。(補足 4.4.C)

例えば以下の操作を行えば良い。

- (1) (作業用フォルダの作成) エクスプローラー  から「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 」フォルダを開く。  
エクスプローラー画面の左上の「新規作成」メニューから「フォルダー」を選択し、生成したフォルダ「新しいフォルダ」の名称を「20240503-021」とする。
- (2) (処理前強度ファイルの設置) エクスプローラー  から「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex」フォルダ中の“20240503-021.csv” ファイルをチェックし、「コピー」アイコン  をクリック (タップ) する。  
「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダを開き、「貼り付け」アイコン  をクリック (タップ) する。  
「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダ中の「 20240503-021.csv」ファイルをチェックし、エクスプローラーの「名前の変更」アイコン  をクリック (タップ) し、名称を「00raw.csv」に変更する。
- (3) (設定ファイルの設置) エクスプローラー  から「 / OneDrive / Documents / 00jikken / 20240503MiniFlex / 20240503-000 /」フォルダ中の「 cntmn.cfg」 「 dct.cfg」 「 xray.cfg」ファイルにチェックをし、「コピー」アイコン  をクリック (タップ) する。  
「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダを開き、「貼り付け」アイコン  をクリック (タップ) する。
- (4) (作業ディレクトリの内容の確認) Notebook  の Home タブで「 OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダを開き「 00raw.csv」 「 cntmn.cfg」 「 dct.cfg」 「 xray.cfg」の4つのファイルのみが含まれていることを確認する。
- (5) (逆畳込的処理の実行) Notebook  の Home タブからの操作で  
「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダに、新規ノートブックを例えば“04extern5.ipynb”として作成する。Notebook のコードセルから  


```
%run ../extern5.7/extern5.py
```

とすれば、「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / extern5.7 /」フォルダに置かれた Python コード extern5.py が起動する。計算終了時に、終了メッセージと計算のために消費された時間を表示させるようにするためには、例えば  

```
1) %%time
2) %run ../extern5.7/extern5.py
3) print("Finished")
```

のようなコードを実行する。 ([補足 4.4.D](#))



- (6) (逆畳込的处理結果の確認) 「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 /」フォルダに置かれた「00raw.csv」強度データに対して逆畳込的处理を施した後のデータ「01dct.csv」の生成されたことを確認する。


逆畳込的处理前後の XRD データを描画するためには、「 / OneDrive - NITech / Documents / 00jikken / 20240503MiniFlex / 20240503-021 / 04extern5.ipynb」の新しいコードセルから (Code 4.4.1) のような Python コードを実行すれば良い。グラフ上の二つの曲線を区別するために凡例 (legend) を付ける必要があることに注意する。

Figure 4.4.1 のようなグラフが表示される。

(Code 4.4.1) 畳込的处理前後の XRD データの描画 (04extern5.ipynb) (↩)

```
1) # 複数ファイル強度の描画
2) import numpy as np
3) import matplotlib.pyplot as plt
4) sRaw = '00raw.csv' # DCT 処理前データファイル名
5) sDCT = '01dct.csv' # 処理後データファイル名
6) # DCT 処理前データファイルの読み込み...
7) source1 = np.loadtxt(sRaw,dtype='float',delimiter=',')
8) x = source1[:,0] # x は各行 0 列目
9) y1 = source1[:,1] # y は各行 1 列目
10) # DCT 処理後データファイルの読み込み...
11) source2 = np.loadtxt(sDCT,dtype='float',delimiter=',')
12) y2 = source2[:,1] # y は各行 1 列目
13) # グラフの描画...
14) fig,ax = plt.subplots(figsize=(6.4,3.2))
15) ax.plot(x,y1,label='raw') # 折れ線グラフ描画, 凡例ラベル指定
16) ax.plot(x,y2,label='DCT') # 折れ線グラフ描画, 凡例ラベル指定
17) ax.xaxis.set_ticks_position('both') # x軸上下に目盛をつける
18) ax.yaxis.set_ticks_position('both') # y軸左右に目盛をつける
19) ax.minorticks_on() # 補助目盛を表示する
20) ax.set_xlabel('$2\Theta$ (^\circ)$') # x軸ラベル設定
21) ax.set_ylabel('Intensity (counts)') # y軸ラベル設定
22) ax.set_xlim(5,140) # x軸描画範囲設定
23) ax.set_ylim(0,) # y軸描画範囲設定
24) plt.legend() # 凡例描画
25) plt.show()
```

(↩)

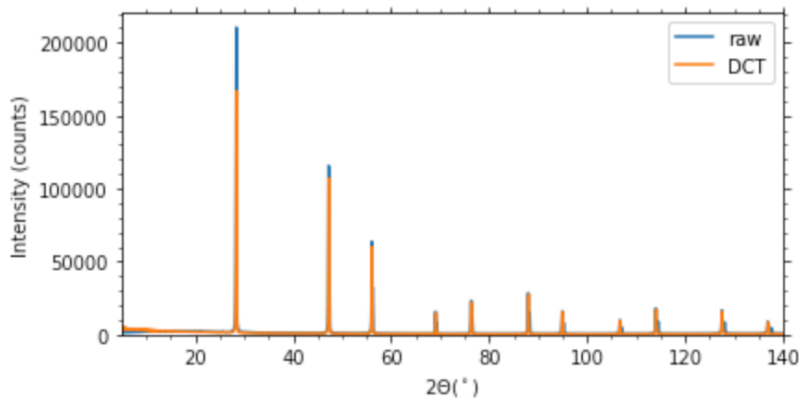


Figure 4.4.1 NIST SRM 640b 実測データ (raw) と逆畳込的処理後データ (DCT) の比較 (↔)

(Code 4.4.1) のうち 22 行目の “ax.set\_xlim(5,140)” を “ax.set\_xlim(110,140)” と変更し、23 行目の “ax.set\_ylim(0,)” を “ax.set\_ylim(0,20000)” と変更すれば、x 軸描画範囲が  $2\theta \in [110^\circ, 140^\circ]$ , y 軸描画範囲が  $y \in [0, 20000]$  と変更され、Figure 4.4.2 に示すようなグラフが得られる。

配布ファイル “20240503MiniFlex” 中に “NIST/SRM640d.pdf” として添付するように、NIST SRM640d の保証書 (certificate) サーティフィケート を見れば Figure 4.4.2 に表示された 3 つのピークは 531, 620, 533 反射であることがわかる。逆畳込的処理前のデータ (raw) では強度比 2:1 の  $\text{CuK}\alpha_1$  主ピークと  $\text{CuK}\alpha_2$  副ピークが現れているが、逆畳込的処理 (DCT) によって  $\text{CuK}\alpha_2$  副ピークのみが取り除かれていることを確認する。

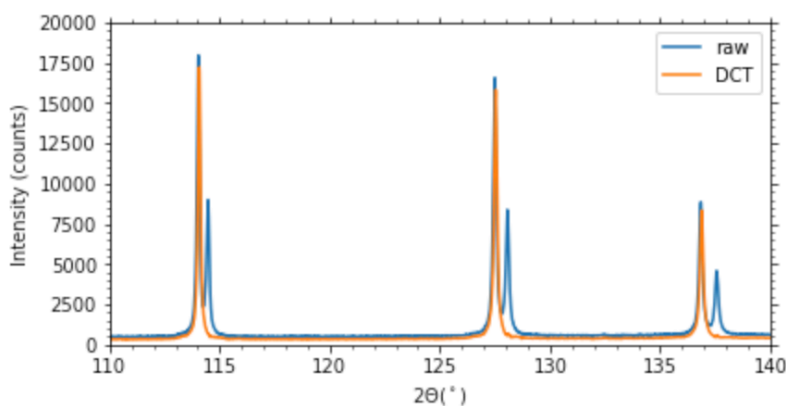


Figure 4.4.2 NIST SRM 640d 実測 XRD データ (raw) と逆畳込的処理後データ (DCT) の比較, 高角領域のデータ (↔)

(Code 4.4.1) のうち x 軸描画範囲を  $2\theta \in [20^\circ, 60^\circ]$ , y 軸描画範囲を  $y \in [0, 4000]$  と変更してコードの実行をし直せば Figure 4.4.3 に示すようなグラフが得られる。

Figure 4.4.3 のグラフに現れる回折ピークは、111, 220, 311 反射である。逆畳込的処理前のデータ (raw) では背景強度に現れる「NiK-吸収端による段構造」と「Ni 箔フィルターにより弱められた  $\text{CuK}\beta$  ピーク」が、逆畳込的処理後のデータ (DCT) では、取り除かれていることを確認する。

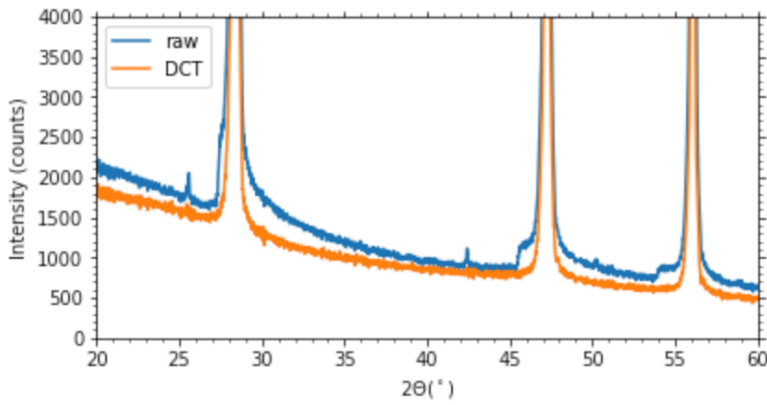


Figure 4.4.3 NIST SRM 640d 実測データ (raw) と逆畳込的处理後データ (DCT) の比較, 低角領域の背景強度データ (↩)

## 補足

### (補足 4.A) NIST 標準試料 (↩)

2023年4月29日の時点では, Si 標準粉末 NIST SRM640 シリーズの最新版である SRM640f (Black et al., [2020a](#)) は, 日本の商社から例えば 178,000 円 / 7.5 g で購入できたが, 2024年3月30日には品切れの状態になった。六硼化ランタン LaB<sub>6</sub> 標準粉末 NIST SRM660 シリーズの最新版である SRM660c (Black et al., [2020b](#)) は 210,000 円 / 6 g という価格設定がされているが, 2024年3月30日の時点では, やはり品切れの状態となった (オーバーシーズ・エックスレイ・サービス <https://www.oxs.co.jp>, 2024年3月30日閲覧)。

**シリコン** Si は単体 (elementary substance) であり, 複数の元素から構成される化合物と異なり, 化学組成が化学量論比からずれる懸念がない。超高純度の原料を比較的容易に入手できる。空間群記号  $Fd\bar{3}m$  (エフ・ディー・スリーバー・エム) (空間群番号 227) の面心立方, ダイヤモンド型構造であり, 近い角度に異なる (等価ではない) 回折ピークの出現することがない。さらに面心立方の複合格子であることとダイヤモンド映進対称 (diamond glide plane symmetry) を持つ空間群対称性から, 出現する反射が限られたものになる。面心立方対称から指数  $hkl$  はすべて奇数かすべて偶数かに限られ, ダイヤモンド映進対称性から  $h+k+l$  の値に対する 4 を除数とする除算の剰余が 2 である反射は禁制反射となる。(補足 4.A.1) 重元素を含まず, 高い回折角のピークの強度は比較的低下する。CuK $\alpha_1$  X線に対する質量減衰係数は  $(\mu/\rho)_{\text{Si}} = 63.7 \text{ cm}^2/\text{g}$ , 計算密度は  $\rho_{\text{calc, Si}} = 2.33 \text{ g/cm}^3$  であり, バルク固体の線減衰係数は  $\mu_{\text{solid, Si}} = 148 \text{ cm}^{-1}$ , X線侵入深さは  $\mu_{\text{solid, Si}}^{-1} = 0.067 \text{ mm}$  である。ゴニオメータ半径  $R = 150 \text{ mm}$ , 発散スリット開き角  $\Phi_{\text{DS}} = 1.25^\circ$ , X線ビーム幅  $W_{\text{X-ray}} = 10 \text{ mm}$  の測定条件の場合, 有効照射体積は

$$\begin{aligned} V_{\text{eff, Si}} &= R \Phi_{\text{DS}} W_{\text{X-ray}} \mu_{\text{solid, Si}}^{-1} \\ &= (150 \text{ mm}) \times \left( 1.25^\circ \times \frac{\pi}{180^\circ} \right) \times (10 \text{ mm}) \times (0.067 \text{ mm}) \approx 2.20 \text{ mm}^3 \end{aligned}$$

と見積もられる。NIST SRM640f の代表的な粒径は  $5.3 \mu\text{m}$  とされており, 有効照射体積中に  $(2.2 \text{ mm}^3)/(5.3 \times 10^{-3} \text{ mm})^3 \approx 1.5 \times 10^7$  個程度の結晶粒の含まれることが期待できる。半導体検出器の画角 (view angle)  $\Psi = 4.89^\circ$ , ソーラーズリット開き角  $\Phi_{\text{SS}} = 1.25^\circ$  の場合, ランダム配向した結晶子が回折条件を満たす確率は

$$p_{\text{Si}} = \left( 4.89^\circ \times \frac{\pi}{180^\circ} \right) \times \left( 1.25^\circ \times \frac{\pi}{180^\circ} \right) \times \frac{1}{2\pi} \approx 3.0 \times 10^{-4}$$

と概算される。ダイヤモンド型構造で反射多重度が最小の 6 となる  $h00$  反射に属する 400 反射であっても  $(1.5 \times 10^7) \times (3.0 \times 10^{-4}) \times 6 \approx 24000$  個程度の結晶粒が観測回折強度に寄与するとみなせる。  $1/\sqrt{24000} \approx 0.006$  から, 観測される回折強度に寄与する粒子数が有限であることによる粉末X線回折ピーク強度の相対的な統計誤差は 0.6% 以下と期待して良い。

Si の場合、最低角度に出現する 111 反射のピーク位置は、波長  $\lambda \approx 1.5406 \text{ \AA}$  の  $\text{CuK}\alpha_1$  X線では  $28.4^\circ$  付近なので、何らかの較正曲線により  $28.4^\circ$  より低い角度の較正をしたとしても、それを正当化することが困難になる面がある。また角度較正に用いる場合、Si 粉末試料の X線に対する透過性に由来するピークシフトの寄与を正しく考慮する必要がある。

**六硼化ランタン**  $\text{LaB}_6$  は二元系の化合物であるから、現実の物質の化学組成が化学量論比からずれる場合がありうることを否定できない。空間群記号  $Pm\bar{3}m$  (空間群番号 221) の立方晶であり、近い位置に異なる  $hkl$  指数反射の現れることはないが、単純立方格子であり、螺旋・映進などの空間群対称性を持たないので、すべての  $hkl$  指数の反射が回折ピークとして出現する。重元素 La を含むので、高い回折角のピークでも比較的高い強度が観測される。 $\text{CuK}\alpha_1$  X線に対する質量減衰係数は  $(\mu/\rho)_{\text{LaB}_6} = 221.8 \text{ cm}^2/\text{g}$ 、計算密度は  $\rho_{\text{calc, LaB}_6} = 4.71 \text{ g/cm}^3$  であり、固体の線減衰係数は  $\mu_{\text{solid, LaB}_6} = 1045 \text{ cm}^{-1}$ 、X線侵入深さは  $\mu_{\text{solid, LaB}_6}^{-1} = 0.010 \text{ mm}$  である。ゴニオメータ半径  $R = 150 \text{ mm}$ 、発散スリット開き角  $\Phi_{\text{DS}} = 1.25^\circ$ 、X線ビーム幅  $W_{\text{X-ray}} = 10 \text{ mm}$  の測定条件の場合、有効照射体積は  $V_{\text{eff, LaB}_6} = (150 \text{ mm}) \times [1.25^\circ \times \pi / (180^\circ)] \times (10 \text{ mm}) \times (0.010 \text{ mm}) \approx 0.33 \text{ mm}^3$  と見積もられる。NIST SRM660c の代表的な粒径は約  $10 \mu\text{m}$  とされており、有効照射体積中に  $3.3 \times 10^5$  個程度の結晶粒の含まれることしか期待できない。半導体検出器の画角  $\Psi = 4.89^\circ$ 、ソーラスリット開き角  $\Phi_{\text{SS}} = 1.25^\circ$  の場合、ランダム配向した結晶子が回折条件を満たす確率は  $3.0 \times 10^{-4}$  と概算される。反射多重度が 6 の 100, 200, ... 反射の場合には約 600 個程度の結晶粒しか観測される回折強度に寄与しうるとみなせない。 $1/\sqrt{600} \approx 0.04$  から、4% 程度の相対的な統計誤差を伴う可能性がある。最低角度に出現する 100-反射のピーク位置は  $21.4^\circ$  付近なので、 $21.4^\circ$  より高い角度領域であれば、較正を正当化することができる。線減衰係数が高い (X線侵入深さが浅い) ので、試料の透過性に由来するピークシフトを無視しても事実上問題にならない。 (↔)

#### (補足 4.A.1) ダイヤモンド構造の消滅則 (↔)

ダイヤモンド構造の空間群対称性は  $Fd\bar{3}m$  であり、面心立方格子であることから、反射指数  $h, k, l$  がすべて奇数か全て偶数の場合にしか回折強度が現れない。このことは、以下のように理解することができる。面心立方構造は A 底心、B 底心、C 底心のすべての対称性を持つ。

A 底心構造では、任意の原子の分率座標  $(x, y, z)$  に対して  $(x, y + 1/2, z + 1/2)$  の位置は等価な原子位置となる。これらの原子によって散乱された散乱波の位相は

$$e^{2\pi i(hx+ky+lz)} + e^{2\pi i[hx+k(y+1/2)+l(z+1/2)]} = e^{2\pi i(hx+ky+lz)} \left[ 1 + e^{2\pi i(k/2+l/2)} \right]$$

と表される。 $e^{\pi i} = -1$  の関係から  $k+l$  の値が偶数の場合には散乱強度が観測されるが、 $k+l$  の値が奇数の場合には散乱強度が観測されない。このことは  $k, l$  がともに奇数であるか、ともに偶数の場合にしか観測強度が観測されないことを意味する。同様に B 底心構造の場合に散乱強度が観測されるのは  $h+l$  の値が偶数である場合に限られ、C 底心構造の場合に散乱強度が観測されるのは  $h+k$  の値が偶数である場合に限られる。面心立方構造では  $h, k, l$  の値がすべて奇数であるかすべて偶数の場合にしか散乱強度が観測されないことになる。

ダイヤモンド構造は、ダイヤモンド映進面 (diamond glide plane) 対称を持つ。この対称性は「 $d$  映進」( $d$ -glide) とも呼ばれる。単位胞の体対角線方向 ( $[111]$  方向) に半周期平行移動して鏡映操作を施すと元の構造と一致するような対称性である。ダイヤモンド映進の対称性について「対角映進面 (diagonal glide plane) 対称」と書かれる例も見られるが、結晶学では「対角映進面对称」は体対角線方向ではなく面对角線方向への平行移動を含む対称操作を意味し「 $n$  映進」( $n$ -glide) と呼ぶ方が標準的である。

ダイヤモンド構造では原子の分率座標  $(x, y, z)$  に対して  $(x + 1/4, y + 1/4, z + 1/4)$  の位置は等価な原子位置となる。これらの原子によって散乱された散乱波の位相は

$$e^{2\pi i(hx+ky+lz)} + e^{2\pi i[h(x+1/4)+k(y+1/4)+l(z+1/4)]} = e^{2\pi i(hx+ky+lz)} \left[ 1 + e^{2\pi i(h+k+l)/4} \right]$$

と表される。 $e^{\pi i} = -1$  の関係から  $h + k + l$  の値が 4 で割って 2 余る数の場合には回折強度が現れない。低回折角（長面間隔）側から高回折角（短い面間隔）側へ向かって、 $\text{CuK}\alpha$  X線源を使った回折装置で観測できるのは、111, 220, 311, 400, 331, 422, 511/333, 440, 531, 620, 533 の 11 本の反射のみである。 (↔)

#### (補足 4.1.A) クラウドストレージでのプログラミング作業 (↔)

クラウドストレージ (cloud storage) を使ったプログラミングやデータ処理の作業は、個人的な利用であってもメリットは多い。インターネット通信のトラブルによって作業性の低下する場合があります、インターネットに接続しない環境では使えないことになるが、最近のシステムでは、それらのことが気になる状況は稀になっている。 (↔)

#### (補足 4.2.A) コンピュータ・プログラミングでのコメント・ドキュメンテーションの重要性 (↔)

Python 言語を用いる場合に限らず、コンピュータ言語を用いた実用的なアプリケーションの開発では、ソース・コードにコメントを付けながらコーディングの作業が行われる。アプリケーションのユーザーにとってはソース・コード中のコメントは意味のないものと思われるかもしれないが、コメントの挿入をせずに実用的で意味のあるアプリケーションがコーディングされることは、現実的にはありえない。

また、実用的なアプリケーションの開発では、コンピュータ言語を用いた「コーディング」の作業より、システムの仕様 (specification) や技術情報、参考文献、取扱説明書 (instruction manual)、初心者向けのチュートリアル、ヘルプ・メッセージ、使用例、デモンストレーションなど「自然言語」で記述されたメッセージ、文書、ドキュメンテーションを作成する作業のために時間も労力も多く費やされる。 (↔)

#### (補足 4.2.B) Jupyter Notebook を使う場合のディレクトリ (フォルダ) の構成 (↔)

市販のアプリケーションプログラムはメモリの占有量が多いので、複数の場所に同じ内容のアプリケーションを置く使い方はしない。それに対して Python スクリプトや Jupyter Notebook のファイルは (一般的には) メモリの占有量が多くないので、データを保存・加工するために利用するコードは、データを保存したサブディレクトリごとに配置する使い方をするのが良い。

データから何らかの情報を抽出したとして、それがどのような処理を施して得られた情報であるかを記録して、後から参照・検証できるようにすることは重要である。記憶の曖昧さから生じる紛らわしさを軽減し、データやコードの保守・再利用・更新・修正などを容易にするメリットがある。

データとコードをまとめておくスタイルをとることは、プログラミング初心者にとっては難度の高い場合の多い「ファイル入出力のためのコーディング」を簡素化できるメリットもある。 (↔)

#### (補足 4.3.A) 繰り返し走査強度データを積算する意味 (↔)

測定に用いた装置 Rigaku MiniFlex 600-C では、 $5^\circ$  付近から  $140^\circ$  付近の範囲を  $10^\circ/\text{min}$  の角速度で走査するのが標準的な使用法と考えて良い。この測定条件でも多くの目的で有効な情報が得られる。この場合、一回の走査測定に概ね 13.5 min の時間がかかる。

しかしこの測定条件では、装置の角度較正の目的に利用できるような十分な統計精度を得ることが難しくなる場合がある。

走査測定強度データの統計精度を高める方法として、一般的に (1) 複数回同一条件での走査測定を繰り返して観測強度の積算をする方法と (2) 走査速度を落とした測定をして、単位時間あたりのカウント数を増やす方法とがある。それぞれの方法には、メリットもデメリットもある。 (↔)



#### (補足 4.4.A) 逆畳込的处理の理論的な基礎 (↔)

ブラッグ プレンターノ

Bragg-Brentano 型の粉末 X 線回折装置によって観測される回折ピークの位置ずれと変形の要因には、主に以下 5 つの要因がある。

(1) **X 線源の分光強度分布 (spectroscopic intensity distribution of source X-ray)** : 銅ターゲットの X 線源と Ni 箔フィルターを用いる場合には、主ピークである  $\text{CuK}\alpha_1$  線の 1/2 の強度を持つ  $\text{CuK}\alpha_2$  線による副ピーク、減衰された  $\text{CuK}\beta$  ピーク、NiK-吸収端に由来する段構造が現れる。これら X 線源の分光強度分布の影響は  $\tan \Theta$  に比例する回折角依存性を示す。

(2) **軸発散収差 (axial-divergence aberration)** : X 線ビーム進行方向のゴニオメータ回転軸と並行な方向 (軸方向) へのずれに由来する装置収差。最適設計である対称ソーラスリット配置では、軸発散収差はソーラスリットの開き角  $\Phi_{\text{SS}}$  によって決まる。軸発散収差の装置関数  $\omega_A(\Delta 2\Theta; 2\Theta, \Phi_{\text{SS}})$  は

$$\omega_A(\Delta 2\Theta; 2\Theta, \Phi_{\text{SS}}) = \frac{1}{\Phi_{\text{SS}}^2} \int_{-\Phi_{\text{SS}}}^{\Phi_{\text{SS}}} \int_{-\Phi_{\text{SS}}}^{\Phi_{\text{SS}}} \delta(\Delta 2\Theta - f_A(\alpha, \beta; 2\Theta)) \left(1 - \frac{|\alpha|}{\Phi_{\text{SS}}}\right) \left(1 - \frac{|\beta|}{\Phi_{\text{SS}}}\right) d\alpha d\beta$$

$$f_A(\alpha, \beta; 2\Theta) = 2\Theta - \arccos(\cos 2\Theta \cos \alpha \cos \beta + \sin \alpha \sin \beta)$$

と表される。軸発散収差による平均ピークシフト  $\Delta 2\Theta_A$  は

$$\Delta 2\Theta_A = \frac{1}{\Phi_{\text{SS}}^2} \int_{-\Phi_{\text{SS}}}^{\Phi_{\text{SS}}} \int_{-\Phi_{\text{SS}}}^{\Phi_{\text{SS}}} f_A(\alpha, \beta; 2\Theta) \left(1 - \frac{|\alpha|}{\Phi_{\text{SS}}}\right) \left(1 - \frac{|\beta|}{\Phi_{\text{SS}}}\right) d\alpha d\beta$$

と表される。2 次近似では軸発散収差による平均ピークシフト  $\Delta 2\Theta_A$  は

$$\Delta 2\Theta_A \approx \frac{\Phi_{\text{SS}}^2}{12} \left( \tan \Theta - \frac{1}{\tan \Theta} \right)$$

と表される。

(3) **赤道収差 (equatorial aberration)** : ゴニオメータ回転軸と垂直な方向 (赤道方向) への X 線ビーム進行方向のずれに由来する装置収差。発散スリット開き角  $\Phi_{\text{DS}}$  とフォトダイオードアレイ (シリコン・ストリップ) 型 X 線検出器の面角  $2\Psi$ , ゴニオメータ半径  $R$ , 試料の幅  $W$  によって決まる。赤道収差の装置関数  $\omega_E(\Delta 2\Theta; 2\Theta, \Phi_{\text{DS}}, 2\Psi)$  は

$$\omega_E(\Delta 2\Theta; 2\Theta, \Phi_{\text{DS}}, 2\Psi) = \frac{1}{\Phi_{\text{DS}} \Psi} \int_{\Phi_{\text{min}}}^{\Phi_{\text{max}}} \int_{-\Psi/2}^{\Psi/2} \delta(\Delta 2\Theta - f_E(\phi, \psi; 2\Theta)) d\psi d\phi$$

$$\Phi_{\text{min}} = \max \left( -\frac{\Phi_{\text{DS}}}{2}, -\frac{\sin \Theta}{\cos \Theta - W/2R} \right)$$

$$\Phi_{\text{max}} = \min \left( \frac{\Phi_{\text{DS}}}{2}, \frac{\sin \Theta}{\cos \Theta + W/2R} \right)$$

$$f_E(\phi, \psi; 2\Theta) = \Theta + \phi + \psi - \arctan \frac{\sin(\Theta + \psi)}{g_E(\phi, \psi; 2\Theta)}$$

$$g_E(\phi, \psi; 2\Theta) = \cos(\Theta - \psi) \cos 2\psi + \cos(\Theta + \psi) - \frac{\sin(\Theta - \psi) \cos 2\psi}{\tan(\Theta - \psi - \phi)}$$

のように表される。赤道収差による平均ピークシフト  $\Delta 2\Theta_E$  は

$$\Delta 2\Theta_E = \frac{1}{\Phi_{\text{DS}} \Psi} \int_{\Phi_{\text{min}}}^{\Phi_{\text{max}}} \int_{-\Psi/2}^{\Psi/2} f_E(\phi, \psi; 2\Theta) d\psi d\phi$$

と表される。2 次近似では、入射ビームが試料からはみ出さない角度領域での赤道収差による平均ピークシフト  $\Delta 2\Theta_E$  は

$$\Delta 2\Theta_E \approx -\frac{\Phi_{DS}^2}{6 \tan \Theta}$$

と表される。

- (4) **試料透過性収差 (sample transparency aberration)** : 試料に対する X 線の透過性に由来する収差。試料の線減衰係数  $\mu$  とゴニオメータ半径  $R$ , 試料の厚さ  $t$  によって決まると考えられていた。試料透過性収差の装置関数  $\omega_{ST}(\Delta 2\Theta; 2\Theta, \mu, R, t)$  は

$$\omega_{ST}(\Delta 2\Theta; 2\Theta, \mu, R, t) = \int_{-t}^0 \delta(\Delta 2\Theta - f_{ST}(z; 2\Theta, R)) g_{ST}(z; 2\Theta, \mu) dz$$

$$f_{ST}(z; 2\Theta, R) = 2\Theta - 2 \arctan \left( \tan \Theta - \frac{z}{R \cos \Theta} \right)$$

$$g_{ST}(z; 2\Theta, \mu) = \frac{2\mu}{\sin \Theta} \exp \left( \frac{2\mu z}{\sin \Theta} \right)$$

のように表される。

試料透過性収差による平均ピークシフト  $\Delta 2\Theta_{ST}$  は

$$\Delta 2\Theta_{ST} = \int_{-t}^0 f_{ST}(z; 2\Theta, R) g_{ST}(z; 2\Theta, \mu) dz$$

と表され、1次近似では試料透過性収差による平均ピークシフト  $\Delta 2\Theta_{ST}$  は

$$\Delta 2\Theta_{ST} \approx -\frac{\sin 2\Theta}{2\mu R} + \left( \frac{\sin 2\Theta}{2\mu R} + \frac{2t \cos \Theta}{R} \right) \exp \left( -\frac{2\mu t}{\sin \Theta} \right)$$

と表される。試料厚さ  $t$  が X 線侵入深さ  $\mu^{-1}$  より十分に大きい場合 ( $t \gg \mu^{-1}$ ) には、1次近似の試料透過性収差による平均ピークシフトは

$$\Delta 2\Theta_{ST} \approx -\frac{\sin 2\Theta}{2\mu R}$$

と表される。

- (5) **試料位置ずれ効果 (specimen displacement effect)** : 試料の表面がゴニオメータの回転軸からずれた位置にあることによる回折ピーク位置のずれ。回折ピーク形状は変化しない。ゴニオメータ半径  $R$  と試料の位置ずれ  $\Delta S$  によって決まる。試料面の上方方向にずれた場合に  $\Delta S$  が正の値をとるとして、ピークの位置のずれは1次近似では

$$\Delta 2\Theta_{SD} \approx \frac{2\Delta S \cos \Theta}{R}$$

と表される。試料位置ずれによる回折ピーク角度のずれは  $\cos \Theta$  に比例して変化する。

2つの関数  $f(x)$  と  $g(x)$  の たたみこみ コンボリューション **畳込 (convolution)** は、通常  $f(x) * g(x)$  と表される。印刷物等では Python などのコンピュータ言語で乗算記号として用いられるアスタリスク「\*」と同じ記号を使って表現されるが、乗算という意味でないことに注意する。TeX を使える環境であれば  $f(x) \otimes g(x)$  ( $f(x) \ \backslash \otimes \ g(x)$ ) のように表現することもできる。

関数  $f(x)$  と  $g(x)$  の畳込は

$$f(x) * g(x) \equiv \int_{-\infty}^{\infty} f(x-y)g(y) dy \tag{4.4.A.1}$$

あるいは

$$f(x) * g(x) \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x-y-z)f(y)g(z) dy dz \tag{4.4.A.2}$$

として定義される。ただし  $\delta(x)$  は **ディラックのデルタ関数 (Dirac delta)** である。

フーリエ  
 畳込の Fourier 変換は成分函数の Fourier 変換の積に等しい。 $h(x) = f(x) * g(x)$  のとき、

$$\mathfrak{H}(k) \equiv \int_{-\infty}^{\infty} h(x) e^{2\pi i k x} dx \quad (4.4.A.3)$$

$$\mathfrak{F}(k) \equiv \int_{-\infty}^{\infty} f(x) e^{2\pi i k x} dx \quad (4.4.A.4)$$

$$\mathfrak{G}(k) \equiv \int_{-\infty}^{\infty} g(x) e^{2\pi i k x} dx \quad (4.4.A.5)$$

とすれば、 $(\mathfrak{H}, \mathfrak{F}, \mathfrak{G})$  はそれぞれ H, F, G のフラクトゥール文字)

$$\mathfrak{H}(k) = \mathfrak{F}(k) \mathfrak{G}(k) \quad (4.4.A.6)$$

の関係が成立する。この関係を<sup>たたみこみていり</sup>畳込定理 (convolution theorem) と呼ぶ。<sup>コンヴォリューション セオレム</sup>

かりに  $h(x)$  が実測の強度データ、 $f(x)$  が装置の影響でシフトや変形をする前の本来の強度データ、 $g(x)$  が装置取差の影響によるピークの位置シフトと変形を表す函数「<sup>インストゥルメンタル ファンクション</sup>装置函数 (instrumental function)」であるとすれば、

$$f(x) = \int_{-\infty}^{\infty} \frac{\mathfrak{H}(k)}{\mathfrak{G}(k)} e^{-2\pi i k x} dk \quad (4.4.A.7)$$

の計算をすることによって「実測の強度データ」と「装置函数」とから「装置の影響を受ける前の本来の強度データ」を求められるはずであるとする考え方を<sup>ぎやくたたみこみ</sup>逆畳込 (deconvolution) と呼ぶ。<sup>デコンボリューション</sup>しかし、現実には逆畳込が有効に機能する場合は稀であり、<sup>ま</sup>逆畳込計算の結果を意図的にぼやけさせる<sup>へいかつかしより</sup>平滑化処理 (smoothing) も同時に施されるのが普通である。

粉末 X 線回折強度図形に現れる X 線の分光強度分布の影響については、ゴニオメータ角度を  $2\Theta$ 、強度値を  $Y$  とする  $(2\Theta, Y)$  データから

$$\chi_X = \ln \sin \Theta \leftarrow 2\Theta \quad (4.4.A.8)$$

$$\eta_X = \frac{Y}{L(2\Theta)} \frac{d2\Theta}{d\chi_X} = \frac{2Y \tan \Theta}{L(2\Theta)} \leftarrow Y \quad (4.4.A.9)$$

$$L(2\Theta) = \frac{1 + \cos^2 2\Theta}{\sin \Theta \sin 2\Theta} \quad (4.4.A.10)$$

のように対数正弦尺度変換 (logarithmic sine scale transform) と幾何学的な補正 (逆補正) を施して  $(\chi_X, \eta_X)$

( $\chi$  はギリシャ文字小文字のカイ; chi) ( $\eta$  はギリシャ文字小文字のイータ; eta) のように変換すれば、実測の XRD データは分光強度分布との畳込として表現される。式 (4.4.A.10) で表される  $L(2\Theta)$  は幾何学的な補正因子である。

分光強度分布については、Deutsch ら (Deutsch et al., 2004) による (1) <sup>しじゅう</sup>CuK $\alpha$  四重線モデルと (2) CuK $\beta$  五重線モデルを減衰させたモデル、また NiK 吸収端については裁断指数函数 (truncated exponential function) :

$$f_{TE}(\chi; \chi_{NiK}, \gamma_{NiK}) = \begin{cases} \exp\left(-\frac{\chi - \chi_{NiK}}{\gamma_{NiK}}\right) & [\chi \geq \chi_{NiK}] \\ 0 & [\chi < \chi_{NiK}] \end{cases} \quad (4.4.A.11)$$

の和としてモデル化する。式 (4.4.A.11) 中の  $\chi_{NiK}$  は NiK 吸収端位置、 $\gamma_{NiK}$  ( $\gamma$  はギリシャ文字小文字のガンマ; gamma) は減衰幅を表す。

分光強度分布モデルについて逆畳込処理を施すのと同時に、CuK $\alpha_1$  位置に CuK $\alpha_1$  線と同程度の<sup>せんはば</sup>線幅  $w_{Lor}$  を持つローレンツ型函数 :

$$f_{Lor}(\chi; \chi_{CuK\alpha_1}, w_{Lor}) = \frac{1}{\pi w_{Lor}} \left[ 1 + \left( \frac{\chi - \chi_{CuK\alpha_1}}{w_{Lor}} \right)^2 \right]^{-1} \quad (4.4.A.12)$$

と、NiK 吸収端の除去に用いた裁断指数函数と同程度の強度と広がりを持つ双曲正割函数

$$f_{\text{sech}}(\chi, \chi_{\text{Cu}K\alpha_1}, w) = \text{sech} \left( \frac{\chi - \chi_{\text{Cu}K\alpha_1}}{w_{\text{sech}}} \right) \quad (4.4.A.13)$$

を畳み込む。

装置収差については、装置収差関数の逆畳込計算と平滑化処理をする代わりに、実測値の Fourier 変換  $\mathfrak{S}(k)$  に装置収差関数の Fourier 変換  $\mathfrak{G}(k)$  との除算を施すのと同時に、装置収差関数の Fourier 変換の複素絶対値  $|\mathfrak{G}(k)|$  との乗算を施してから逆 Fourier 変換を施すことを提案する。つまり、

$$f(x) = \int_{-\infty}^{\infty} \frac{\mathfrak{S}(k) |\mathfrak{G}(k)|}{\mathfrak{G}(k)} e^{-2\pi i k x} dk \quad (4.4.A.14)$$

の計算を行う。このことを ぎやくたみこみてき **逆畳込的処理 (deconvolutional treatment)** と呼ぶ。

逆畳込的処理は、装置収差関数  $g(x)$  の偶数階数キュムラント (cumulant) の影響は変化させず、奇数階数キュムラントの影響を無効化する。函数  $g(x)$  の  $k$  階キュムラント  $\kappa_k$  ( $\kappa$  はギリシャ文字小文字の kappa) は、

$$\kappa_k \equiv \left[ \frac{\partial^k}{\partial t^k} \ln \int_{-\infty}^{\infty} e^{tx} g(x) dx \right]_{t=0} \quad (4.4.A.15)$$

として定義される。

装置収差に関する逆畳込的処理は、0 階のキュムラントに対応付けられる積分強度、2 階のキュムラントに対応付けられる線幅広がり、4 階のキュムラントに対応付けられる「ピーク形状の尖り」は変化させず、1 階のキュムラントに対応付けられるピークシフト、3 階のキュムラントに対応付けられる「ピーク形状の非対称な変形」を修正する。結果として「本来の位置」に左右対称なピーク形状が現れることが期待されることになる。

装置収差関数が

$$\omega(\Delta 2\Theta; \alpha_{\min}, \alpha_{\max}, \beta_{\min}, \beta_{\max}) = \int_{\alpha_{\min}}^{\alpha_{\max}} \int_{\beta_{\min}}^{\beta_{\max}} \delta(\Delta 2\Theta - f(\alpha, \beta)) g(\alpha, \beta) d\beta d\alpha \quad (4.4.A.16)$$

の形式で表される場合、 $\Delta 2\Theta$  の  $k$  階<sup>べき</sup>平均  $\langle (\Delta 2\Theta)^k \rangle$  は

$$\langle (\Delta 2\Theta)^k \rangle = \int_{\alpha_{\min}}^{\alpha_{\max}} \int_{\beta_{\min}}^{\beta_{\max}} [f(\alpha, \beta)]^k g(\alpha, \beta) d\beta d\alpha \quad (4.4.A.17)$$

として計算できるので、装置収差関数そのものを求めることが難しい場合でも、装置収差関数の 4 階までのキュムラントを求めることは難しくない。(補足 4.4.A.1)

畳込のキュムラントは成分函数のキュムラントの和に等しい。この関係を「畳込におけるキュムラントの可加算性」(additivity of cumulants on convolution) と呼ぶ。この関係から多重の畳込のキュムラントを求めることも容易である。

(↔)

#### (補足 4.4.A.1) 低階<sup>べき</sup>のキュムラント (↔)

強度曲線を表す函数  $f(x)$  について、変数  $x$  の  $k$  階<sup>べき</sup>積分  $s_k$  を

$$s_k = \int_{-\infty}^{\infty} x^k f(x) dx \quad (4.4.A.1.1)$$

とすれば、積分強度  $s_0$  と  $k$  階<sup>べき</sup>キュムラント  $\kappa_k$  は

$$\kappa_0 = \ln s_0, \quad (4.4.A.1.2)$$

$$\kappa_1 = \frac{s_1}{s_0}, \quad (4.4.A.1.3)$$

$$\kappa_2 = \frac{s_2}{s_0^2} - \frac{s_1^2}{s_0^2}, \quad (4.4.A.1.4)$$

$$\kappa_3 = \frac{s_3}{s_0^3} - \frac{3s_2s_1}{s_0^3} + \frac{2s_1^3}{s_0^3}, \quad (4.4.A.1.5)$$

$$\kappa_4 = \frac{s_4}{s_0^4} - \frac{4s_3s_1}{s_0^4} - \frac{3s_2^2}{s_0^4} + \frac{12s_2s_1^2}{s_0^4} - \frac{6s_1^4}{s_0^4} \quad (4.4.A.1.6)$$

などのように表される。 (↔)

#### (補足 4.4.B) 逆畳込的処理のための Python コード (↔)

逆畳込的処理を実施するための 12 ファイルからなる Python コードは、それぞれ以下の機能を持つ。

- (1) “axial5.py”: 装置の軸発散収差 (axial-divergence aberration) の修正 (13 kB)
- (2) “cntmn5.py”: 当初は X 線発生部である銅 Cu ターゲットへのタングステン W 蒸着による汚染 (contamination) に対応することを意図して開発したが、X 線源中の Ni 合金部材から発生する X 線によると推測される偽ピークを除去する目的に用途を変更した。パナリティカル (Panalytical) (旧フィリップス Philips) 社製の X 線源を用いるときに必要となる場合がある。キャノン電子管デバイス (旧東芝真空管) の X 線源を用いるときに必要となる場合は確認していない。(24 kB)
- (3) “common5.py”: 逆畳込的処理 (deconvolutional treatment) で共通して (commonly) 用いるルーチンをまとめたもの (10 kB)
- (4) “equatorial1.py”: 装置の赤道収差 (equatorial aberration) の修正 (1) (24 kB)
- (5) “equatorial2.py”: 装置の赤道収差 (equatorial aberration) の修正 (2) (15 kB)
- (6) “equatorial3.py”: 装置の赤道収差 (equatorial aberration) の修正 (3) (15 kB)
- (7) “extern5.py”: 逆畳込的処理の主プログラム (19 kB)
- (8) “transpr1.py”: 試料透過性収差 (sample transparency aberration) の修正。繰り返し計算アルゴリズムを利用する場合 (17 kB)
- (9) “transpr2.py”: 試料透過性収差 (sample transparency aberration) の修正。安直二段階逆畳込的法 (naive two-step deconvolutional method) を利用する場合 (17 kB)
- (10) “transpr3.py”: 不透明な (opaque) 試料ホルダを用いる場合の試料透過性収差 (sample transparency aberration) の修正。安直でない二段階逆畳込的法 (non-naive two-step deconvolutional method) を利用する (17 kB)
- (11) “transpr4.py”: 半透明な (translucent) 試料ホルダを用いる場合の試料透過性収差 (sample transparency aberration) の修正。安直でない二段階逆畳込的法 (non-naive two-step deconvolutional method) を利用する (17 kB)
- (12) “xray5.py”: X 線源の分光強度分布の影響の処理 (15 kB)

(↔)

#### (補足 4.4.C) 逆畳込的処理のための設定ファイル dct.cfg (↔)

逆畳込的処理アプリケーション extern5.7 を利用するためには、測定装置のデザインと試料の性状、測定条件に適合させた cntmn.cfg, dct.cfg, xray.cfg の 3 つの設定ファイルを用いる。キャノン電子管デバイス (Canon Electron Tubes & Devices) 社 A-21 Cu を X 線源とし、公称厚さ 0.023 mm の Ni 箔フィルターを用いた XRD 測定結果を処理するためには dct.cfg の内容のみを書き換えれば良い。

以下のような情報が必要となる。

- (1) 装置のデザイン。ブラッグ・ブレンターノ (Bragg-Brentano) 型以外の場合には対応しない。



- (2) ゴニオメータ半径 (goniometer radius) (カメラ長)  $R$
- (3) 発散スリット開き角 (divergence-slit open angle)  $\Phi_{DS}$
- (4) 対称配置ソーラースリット (Soller slits) 角 (ソーラースリット中の金属箔の間隔と長さの比の逆正接 アークトアンジェント  $\arctangent$ )  $\Phi_{SS}$ 。または、非対称配置ソーラースリットの場合の入射側ソーラースリット角  $\Phi_{SS}^{(i)}$  と回折側ソーラースリット角  $\Phi_{SS}^{(d)}$ 。日本に本拠地を置く XRD 装置製造会社であるリガク社では  $2\Phi_{SS}$  に相当する値を「ソーラースリット角」と称するやや例外的な慣習があることに注意する。
- (5) フォトダイオードアレイ型 (ストリップ stripe 型) X線検出器の画角 (view angle)  $2\Psi$  (ツーサイ two psi) または検出ストリップの間隔  $\Delta s_{strip}$  と検出ストリップの本数  $N_{strip}$ 、ゴニオメータ半径  $R$ 。  $2\Psi = 2 \arctan \frac{N_{strip} \Delta s_{strip}}{2R}$  の関係がある。
- (6) X線源の種類。キャノン電子管デバイス  $CuK\alpha$  線源以外のX線源以外の場合には注意が必要である。 ([補足 4.4.C.1](#))
- (7)  $K\beta$  線減衰フィルタの種類と厚さ。 $CuK\alpha$  線源で厚さ 0.023 mm の Ni 箔フィルタを用いる場合と異なるフィルタを用いる場合には xray.cfg の書き換えが必要になる。
- (8) 回折光学素子 (分光器 (monochromator) ・結晶分析器 (crystal analyzer) ・高配向熱分解黒鉛 (highly-oriented pyrolytic graphite; HOPG) 分析器 ・人工超格子多層膜鏡 (ゲーベルミラー Goebel mirror) ) あるいは高分解能パルス高分析器 (pulse-height analyzer; PHA) の使用の有無。これらのいずれかを装備した装置から出力されるデータには、原則的に対応しない。
- (9) 試料粉末の線X線減衰係数  $\mu$  , あるいは化学組成と嵩密度 (かさ bulk density) ([補足 4.4.C.2](#))
- (10) 半透明試料ホルダを用いる場合には試料ホルダの線X線減衰係数  $\mu'$
- (11) 粉末試料の有効幅  $W$
- (12) 粉末試料の有効厚さ  $t$

これらの情報に適合するように設定ファイル“dct.cfg”を書き換えてから利用する。

本演習で用いる設定ファイル“dct.cfg”の内容を以下に示す。

- ```

26) #-*- coding: utf-8 -*-
27) # dct.cfg
28) # configuration file for deconvolver "extern5"
29)
30) # Sample: Glycine, powder
31) #####
32) # Control parameters except X-ray & contamination
33) # X-ray parameters are in xray.cfg
34) # contamination parameters are in cntmn.cfg
35) #####
36)
37) [Control]
38) # Select error evaluation model
39) # 0: no error evaluation, other type has been implemented but not tested
40) error_type = 0
41)
42) # Line numbers of header part in the data file
43) # headerLines = 424 for full header setting of Rigaku SmartLab Studio II

```

```

44) headerLines = 0
45)
46) # Skip any process (skip... = 1)
47) # Output cumulant file (skip... = 2)
48) skipXray = 0
49) skipAxial = 2
50) skipEquat = 2
51) skipTrans = 2
52) skipContamination = 1
53)
54) # Angular error (used in error estimation) (deg.)
55) # sqrt( ( (0.1/185)**2 + (0.15/185)**2 ) / 12 ) * 180 / PI = 0.0161
56) # sqrt( ( (0.1/150)**2 + (0.1/150)**2 ) / 12 ) * 180 / PI = 0.0156
57) # sqrt(0.0156^2+0.01^2) = 0.0185
58) hrz_error = 0.0185
59)
60) # Margin parameter for deconvolutional treatment of spectroscopic profile
61) marginXray = 10.0
62)
63) # Axial divergence angle (deg.) (used for axial-divergence correction)
64) degAxial = 1.25
65) degAxial2 = 1.18
66) # Shape parameter for axial model
67) alphaAxial = 0.70
68) # Margin parameter for deconvolutional treatment of axial-divergence effect
69) marginAxial = 10.0
70)
71) # Divergence Slit Open Angle (deg.) (used for correction of equatorial aberration)
72) # degDS = 1.25
73) degDS = 1.25
74) # View angle (deg.) of Si strip x-ray detector
75) # 2 arctan((number of strips)*(strip interval)/(2 gonioR))
76) degLPSD = 4.89
77) # gonioR: goniometer radius
78) gonioR = 150
79) # specW: specimen width (mm)
80) specW = 20.0
81) # specS: specimen shape
82) specS = rect
83) # beamW: X-ray beam width (mm)
84) beamW = 10
85) # Shape parameter for equatorial model
86) alphaEquat = 0.50
87) # Margin parameter for deconvolutional treatment of equatorial aberration
88) marginEquat = 10.0

```

```

89) # Select algorithm (modeEquat = 1 or 2) for equatorial aberration
90) # 1: Naive method [Ida, T. (2021)]
91) # 2: Non naive method for finite width [Ida, T. (2023)]
92) modeEquat = 1
93)
94) # muInv: penetration depth (mm)
95) muInv = 0.218
96) # muInv2: penetration depth of holder (mm)
97) muInv2 = 0.138
98) # specT: specimen thickness (mm)
99) specT = 0.618
100) # Margin parameter for deconvolutional treatment of sample transparency aberration
101) marginTrans = 10.0
102) # Select algorithm (modeTrans = 1, 2 or 3) for sample-transparency
103) # 1: Iterative method for finite thickness [Ida, T. (2021)]
104) # 2: Naive method for finite thickness and width [Ida, T. (2022)]
105) # 3: Non naive method for finite thickness and width [Ida, T. (2023)]
106) # 4: Non naive method for finite thickness and width, translucent holder [Ida, T. (2024)]
107) modeTrans = 4
108) # nzTrans: sampling points of numerical integral along Z
109) nzTrans = 20
110) # nxTrans: sampling points of numerical integral along X
111) nxTrans = 20

```

同じ装置・測定条件で異なる試料について測定をした場合には、95行目の試料のX線侵入深さ  $\mu\text{Inv}$  ( $\mu$  inverse;  $\mu^{-1}$ ) と 97行目の試料ホルダのX線侵入深さ、99行目の  $\text{specT}$  (specimen thickness;  $t$ ) を変更する。

(↔)

#### (補足 4.4.C.1) 逆畳込的処理のための設定ファイル `cntmn.cfg` と `xray.cfg` (↔)

Python コード `cntmn.py` とそのための設定ファイル `cntmn.cfg` (configuration file for contamination) は、逆畳込的処理ソフトウェア `extern` シリーズを開発する初期段階では、銅ターゲットX線源のタングステン汚染に対応する意図であったが、その後パナリティカル (Panalytical) (旧フィリップス Philips) 社の製造したX線源中の Ni 合金部材から発生すると推測される偽<sup>にせ</sup>ピークを除去する目的に用途を変更した。

銅ターゲットのタングステン汚染により出現する  $W\text{L}$  ピークには `xray.cfg` を書き換えることで対応できる。パナリティカル (旧フィリップス Philips) 社製のX線源を用いる場合には `cntmn.cfg` を書き換えることの必要となる場合がある。キャノン電子管デバイス (旧東芝真空管) 社の製造したX線源を使用する場合に `cntmn.cfg` の書き換えの必要になる事例は確認されていない。

このことはパナリティカル社の製品よりキャノン電子管デバイス社の製品の方が優れているということを意味するわけではない。精密な測定を目的とする場合には、X線発生部付近の温度上昇により金属部材が熱膨張することによって、発光点位置のずれることが懸念点になる。装置設計者の立場からは、熱膨張係数の小さい Ni 合金部材をX線発生装置で効果的に利用することは当然の発想であると言える。

銅以外のターゲットを用いたX線源を用いる場合や、厚さ 0.023 mm 以外の Ni 箔フィルターを用いる場合であっても、`xray.cfg` を書き換えれば対応しうる。(↔)

## (補足 4.4.C.2) 化学組成と高密度から X 線減衰係数を求める方法 (↔)

粉末 X 線回折測定に使用した X 線が  $\text{CuK}\alpha$  線であるとする。X 線の波長は  $\lambda = 1.5406 \text{ \AA}$  としても良い。粉末 X 線回折データを解析するための X 線減衰係数の見積には高い精度は必要とされない。

X 線の光子エネルギー (photon energy)  $h\nu$  を電子ボルト (eV) 単位で表現した値を求める。このためには、

$$eV = h\nu = \frac{hc}{\lambda} \Rightarrow V = \frac{hc}{e\lambda} \quad (4.4.C.2.1)$$

の関係をを用いる。ここで  $e \equiv 1.60217663 \times 10^{-19} \text{ C}$  は素電荷、 $h \equiv 6.62607015 \times 10^{-34} \text{ J s}$  はプランク定数、 $c \equiv 299792458 \text{ m s}^{-1}$  は真空中の光速を表す。また  $1 \text{ \AA} = 10^{-10} \text{ m}$  の関係をを用いる。

SciPy から基礎的な物理定数の数値を参照することができる。Jupyter Notebook を利用して  $\text{CuK}\alpha$  X 線の光子エネルギーを求めるためには (Code 4.4.C.2.1) に示すようなコードを走らせれば良い。

(Code 4.4.C.2.1) 指定波長の光子エネルギー (↔)

```
from scipy.constants import *
wL = 1.5406E-10 # wave length of X-ray in [m]
eV = h*c/(e*wL) # X-ray photon energy in [eV]
print(eV)
```

(↔)

(Code 4.4.C.2.1) を走らせれば 8047.786 ... と数値が表示されることから、波長  $1.5406 \text{ \AA}$  の X 線について光子エネルギーは 8.048 keV あるいは 8.05 keV とみなせることがわかる。

任意の化学組成を持つ化合物の X 線質量減衰係数 (X-ray mass attenuation coefficient) を求めるためには、”20240503MiniFlex/xmac3” フォルダの中に置いた Python コード “xmac3.py” を用いられたい。この Python コードは担当者の web サイト ([https://takashiida.net/codes/code-x-ray\\_mass\\_attenuation\\_coefficient/](https://takashiida.net/codes/code-x-ray_mass_attenuation_coefficient/)) から公開している。2451 行にわたる長大な Python コードであるが、その内容の大部分は、米国国立標準技術研究所が web サイトから NIST Standard Reference Database (SRD) 126 (2004) として公開されている「元素ごとの質量減衰係数」(Table 3) を参照して、自動的に Python コードを生成する機能を持つ別の自作 Python コードによって作成したものである。

この Python コード xmac3.py を用いて得た質量減衰係数に基づく内容を公開するためには、例えば本文中で

「X 線波長  $\lambda = 1.5406 \text{ \AA}$  での質量減衰係数は、NIST Standard Reference Database 126 (2004) に基づいて見積もった。」

などと引用 (citation) をし、参考文献 (reference) の節に

NIST Standard Reference Database 126 (2004). URL: <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients> (最終閲覧 2023 年 5 月 1 日)

と記載するなど、適切な引用を行わなければならない。

Jupyter Notebook での作業用フォルダのサブフォルダ “xmac3” に “xmac3.py” が置かれている場合に、 $\text{CuK}\alpha_1$  特性 X 線 (波長  $1.5406 \text{ \AA}$ ) に対する六硼化ランタン  $\text{LaB}_6$  の質量減衰係数を求めたければ、Notebook のコードセルから

```
%run xmac3/xmac3.py 1.5406 LaB6
```

のようなコードを実行すれば良い。

以下、Python コードの利用などに特別な難しさを感じる人にとっては、参考になるかもしれない情報を記す。

NIST SRD126 (2004) の Table 3 では、例えばランタン (Lanthanum) La と硼素 (Boron) B の質量減衰係数 (mass attenuation coefficient) と質量エネルギー吸収係数 (質量吸収係数) (mass energy-absorption coefficient) について、Table 4.4.C.2.1 と Table 4.4.C.2.2 のような表が示される。質量減衰係数は  $\mu/\rho$  ( $\mu$  over  $\rho$ ) の記号で表され、質量吸収係数は  $\mu_{\text{en}}/\rho$  の記号で表される。X 線が物質を透過する際の減衰の大部分はエネルギー移動をとともなう吸収 (absorption) あるいは非弾性散乱 (inelastic scattering) によるが、一部はエネルギー移動をとともなわない弾性散乱 (elastic scattering) によっても起こされることから、粉末 X 線回折実験などで X 線の減衰を考慮した解析を行うためには、質量吸収係数でなく質量減衰係数を用いるのが正しい。

Table 4.4.C.2.1 ランタン La の X線質量減衰係数  $\mu$  と質量吸収係数  $\mu_{\text{en}}$  (一部) (↔)

|    | Energy<br>(MeV) | $\mu/\rho$<br>(cm <sup>2</sup> /g) | $\mu_{\text{en}}/\rho$<br>(cm <sup>2</sup> /g) |
|----|-----------------|------------------------------------|------------------------------------------------|
|    | 6.26630E-03     | 5.654E+02                          | 5.221E+02                                      |
| L1 | 6.26630E-03     | 6.518E+02                          | 6.003E+02                                      |
|    | 8.00000E-03     | 3.529E+02                          | 3.290E+02                                      |
|    | 1.00000E-02     | 1.967E+02                          | 1.847E+02                                      |

Table 4.4.C.2.2 硼素 B の X線質量減衰係数  $\mu$  と質量吸収係数  $\mu_{\text{en}}$  (一部) (↔)

|  | Energy<br>(MeV) | $\mu/\rho$<br>(cm <sup>2</sup> /g) | $\mu_{\text{en}}/\rho$<br>(cm <sup>2</sup> /g) |
|--|-----------------|------------------------------------|------------------------------------------------|
|  | 8.00000E-03     | 2.346E+00                          | 2.072E+00                                      |
|  | 1.00000E-02     | 1.255E+00                          | 1.006E+00                                      |

Table 4.4.C.2.1 に示されるように、CuK $\alpha$  X線の光子エネルギー 8.05 keV はランタン La の L1 吸収端エネルギー 6.2663 keV に近いが、8 keV から 10 keV の間には吸収端が存在しない。ランタン La についても硼素 (Boron) B についても、光子エネルギー 8.05 keV での質量減衰係数は 8 keV での値と 10 keV での値の補間 (内挿) (interpolation) によって求められる。

質量減衰係数は光子エネルギーに対して概ね指数函数的に変化するので、元素 X の「光子エネルギー 8.05 keV での質量減衰係数」 $(\mu/\rho)_X(8.05 \text{ keV})$  を求めるためには、8 keV での値  $(\mu/\rho)_X(8 \text{ keV})$  と 10 keV での値  $(\mu/\rho)_X(10 \text{ keV})$  とから、重み付きの相乗平均 (weighted geometric mean) :

$$(\mu/\rho)_X(8.05 \text{ keV}) = [(\mu/\rho)_X(8 \text{ keV})]^{\frac{1.95}{2}} [(\mu/\rho)_X(10 \text{ keV})]^{\frac{0.05}{2}} \quad (4.4.C.2.2)$$

として見積もる。

Jupyter Notebook を用いて式 (4.4.C.2.2) のような数式に沿った計算をする場合には、(Code 4.4.C.2.2) のような Python コードを走らせれば良い。

(Code 4.4.C.2.2) 8.05 keV X線に対する La と B の質量減衰係数 (↔)

```
print('Mass attenuation coefficients for Cu K $\alpha$ ')
print(' La: ', 352.9**((1.95/2))*196.7**((0.05/2)))
print(' B: ', 2.346**((1.95/2))*1.255**((0.05/2)))
(↔)
```

Jupyter Notebook から (Code 4.4.C.2.2) を走らせれば、ランタン La の質量減衰係数は  $(\mu/\rho)_{\text{La}} = 347.8 \text{ cm}^2/\text{g}$ 、硼素 (Boron) B の質量減衰係数は  $(\mu/\rho)_{\text{B}} = 2.310 \text{ cm}^2/\text{g}$  と見積もられることがわかる。

ランタン La の原子量を  $m_{\text{La}} = 138.9$ 、硼素 B の原子量を  $m_{\text{B}} = 10.8$  とすれば、六硼化ランタン LaB<sub>6</sub> の CuK $\alpha$  X線に対する質量減衰係数は

$$(\mu/\rho)_{\text{LaB}_6} = \frac{m_{\text{La}}(\mu/\rho)_{\text{La}} + 6 m_{\text{B}}(\mu/\rho)_{\text{B}}}{m_{\text{La}} + 6 m_{\text{B}}} \quad (4.4.C.2.3)$$

として見積もられる。

Jupyter Notebook を用いて式 (4.4.C.2.3) に示すような計算をする場合には、(Code 4.4.C.2.3) に示すような Python コードを走らせれば良い。



```
(Code 4.4.C.2.3) 8.05 keV X線に対する LaB6 の質量減衰係数 (↩)
print('Mass attenuation coefficients of La, B, LaB6 for Cu Kα')
m_La, n_La = 138.9, 1 # atomic weight, number of atoms for La
m_B, n_B = 10.8, 6 # atomic weight, number of atoms for B
# 8.05 keV での La と B の質量減衰係数の計算...
mu_over_rho_La = 352.9** (1.95/2) * 196.7** (0.05/2)
mu_over_rho_B = 2.346** (1.95/2) * 1.255** (0.05/2)
# LaB6 の質量減衰係数の計算...
mu_over_rho_LaB6 = n_La * m_La * mu_over_rho_La
mu_over_rho_LaB6 += n_B * m_B * mu_over_rho_B
mu_over_rho_LaB6 /= n_La * m_La + n_B * m_B
print(' La: ', mu_over_rho_La)
print(' B: ', mu_over_rho_B)
print(' LaB6: ', mu_over_rho_LaB6)
(↩)
```

Jupyter Notebook から (Code 4.4.C.2.2) を走らせれば、LaB<sub>6</sub> の質量減衰係数は  $(\mu/\rho)_{\text{LaB}_6} = 237.9 \text{ cm}^2/\text{g}$  と見積もられることがわかる。(補足 4.4.C.2.1)

質量減衰係数 (mass attenuation coefficient)  $(\mu/\rho)_{\text{LaB}_6}$  に粉末の嵩密度 (bulk density)  $\rho_{\text{bulk}}$  をかければ線減衰係数 (linear attenuation coefficient)  $\mu$  が求まる。

測定に用いた LaB<sub>6</sub> 粉末の嵩密度が  $\rho_{\text{bulk}} = 1.18 \text{ g/cm}^3$  であったとすれば、X線の線減衰係数 (linear attenuation coefficient)  $\mu$  は  $\mu = (\mu/\rho)_{\text{LaB}_6} \rho_{\text{bulk}} = 280.7 \text{ cm}^{-1}$  と見積もられ、X線侵入深さ (penetration depth) は  $\mu^{-1} = 0.036 \text{ mm}$  と見積もられる。(↩)

#### (補足 4.4.C.2.1) 任意の化学式からX線減衰係数を求めること (↩)

(補足 4.4.C.2) で、web ブラウザを用いて NIST Standard Reference Database (SRD) 126 (2004) から比較的信頼性の高い質量減衰係数の情報を得られることを示した。

NIST SRD 126 では、原子番号 1 番の水素 (Hydrogen) H から、原子番号 92 番のウラン (Uranium) U までの X線質量減衰係数が、[“https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z01.html”](https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z01.html) から [“https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z92.html”](https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z92.html) まで、規則的な web ページ名称 (uniform resource locator; URL) によって提供されているので、インターネットから自動的にデータを収集する (scrape) する Python コードを制作することは困難でない。

NIST SRD 126 のデータに基づいて、X線の波長 (Å 単位) と化学式を入力すれば任意の化合物の X線減衰係数を計算する Python コードを xmac3/xmac3.py として用意した。ただし括弧付きの化学式には対応しない。

Jupyter Notebook で xmac3.py を配置した xmac3 ディレクトリの親ディレクトリで作業する場合にコードセルに

```
%run ../xmac3/xmac3.py 1.5406 LaB6
```

と入力して走らせれば、(Output 4.4.C.2.1.1) に示すような出力が得られる。

```
(Output 4.4.C.2.1.1) xmac3.py の出力 (↩)
#-----
# X-ray Mass Attenuation Coefficient & Mass Absorption Coefficient
# based on NIST Standard Reference Database 126
# URL https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients
# (coded by Takashi Ida, May 3, 2023)
# (updated by Takashi Ida, Oct. 25, 2023)
#-----
Chemical formula: LaB6
Wavelength: 1.5406 Å
Photon energy: 8.048 keV
-----
Element symbol: La, Number of atoms: 1
```

Atomic number: 57, Atomic molar mass: 138.91 g/mol  
Mass attenuation coefficient at 8.0 keV: 352.9 cm<sup>2</sup>/g  
Mass attenuation coefficient at 10.0 keV: 196.7 cm<sup>2</sup>/g  
Mass attenuation coefficient at 8.048 keV: 348.006 cm<sup>2</sup>/g  
Mass absorption coefficient at 8.0 keV: 329.0 cm<sup>2</sup>/g  
Mass absorption coefficient at 10.0 keV: 184.7 cm<sup>2</sup>/g  
Mass absorption coefficient at 8.048 keV: 324.493 cm<sup>2</sup>/g  
-----

Element symbol: B, Number of atoms: 6  
Atomic number: 5, Atomic molar mass: 10.81 g/mol  
Mass attenuation coefficient at 8.0 keV: 2.346 cm<sup>2</sup>/g  
Mass attenuation coefficient at 10.0 keV: 1.255 cm<sup>2</sup>/g  
Mass attenuation coefficient at 8.048 keV: 2.311 cm<sup>2</sup>/g  
Mass absorption coefficient at 8.0 keV: 2.072 cm<sup>2</sup>/g  
Mass absorption coefficient at 10.0 keV: 1.006 cm<sup>2</sup>/g  
Mass absorption coefficient at 8.048 keV: 2.037 cm<sup>2</sup>/g  
-----

Chemical formula: LaB6  
Formula molar mass: 203.770 g/mol  
Mass attenuation coefficient at 8.048 keV: 237.971 cm<sup>2</sup>/g  
Mass absorption coefficient at 8.048 keV: 221.855 cm<sup>2</sup>/g

(↩)

#### (補足 4.4.D) Jupyter Notebook のマジックコマンド (↩)

Jupyter Notebook のコードで % 記号で始まる行は Python 言語のコードではなく、Jupyter Notebook (iPython) 特有のコマンドであり、マジックコマンド (magic commands) と呼ばれる。そのうち “%run ...” のように % 記号一つで始まるものはラインマジック (line magic) と呼ばれ、1 行で動作が完了する。“%%time” のように二つの % 記号で始まるものはセルマジック (cell magic) と呼ばれ、コードセルの内容全体が操作の対象となる。

“%run xxx.py” は、ファイル名 “xxx.py” の Python コードを実行するラインマジックであり、“%%time” はコードセルの実行にかかった時間を表示するセルマジックである。(↩)

## 参考文献

Black, D. R., Mendenhall, M. H., Henins, A., Filliben, J. & Cline, P. (2020a). “Certification of SRM 640f line position and line shape standard for powder diffraction”, *Powder Diffraction* **35**, 156–150.

DOI: <https://doi.org/10.1017/S0885715620000366> (↩)

Black, D. R., Mendenhall, M. H., Brown, C. M., Henins, A., Filliben, J. & Cline, J. P. (2020b).

“Certification of standard reference material 660c for powder diffraction”, *Powder Diffraction* **35**, 17–22.

DOI: <https://doi.org/10.1017/S0885715620000068> (↩)

Deutsch, M., Förster, E., Hölzer, G., Härtwig, J., Härmäläinen, K., Kao, C.-C., Huotari, S. & Diamant, R.

(2004). “X-ray spectrometry of copper: new results on an old subject,” *J. Res. Natl. Inst. Stand. Technol.*

**109**, 75–98. DOI: <https://doi.org/10.6028/jres.109.006> (↩)

Ida, T. (2020). “Application of deconvolutional treatment to powder diffraction data collected with a

Bragg-Brentano diffractometer with a contaminated Cu target and a Ni filter,” *Powder Diffr.* **35**, 166–177.

DOI: <https://doi.org/10.1017/S0885715620000445> (↩)

Ida, T. (2021). “Equatorial aberration for powder diffraction data collected by continuous scan of a silicon strip X-ray detector,” Powder Diffr. **36**, 159-175.

DOI: <https://doi.org/10.1017/S0885715621000403> (↔)

NIST Standard Reference Database 126 (2004). URL: <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients>, accessed May 1, 2023 (↔)